

- **SynaptiCAD Allproducts V10.19b_電子設計自動化(英文)**
軟體名稱: SynaptiCAD Allproducts V10.19b (電子設計自動化)
軟體版本: v10.19
軟體語言: 英文
軟體大小: 65M
運行環境: All Windows
軟體類型: 系統工具
軟體簡介:
SynaptiCAD公司出品的EDA（電子設計自動化）工具集合，包括HDL輸入、仿真、測試、波形分析、管腳映射工具等等,是一個很完整的EDA方案。
包含了9種功能強大的工具：TestBencher Pro；BugHunter Pro；VeriLogger Pro；Waveformer Pro；SimuTAG；PinPort；DataSheet Pro；Timing Diagrammer Pro；Gigawave Viewer。

1.TestBencher Pro

- 用於從獨立於語言的時序邏輯表生成交互式VHDL,Verilog, OpenVera, 和 TestBuilder 測試基準及總線功能的模型。生成的測試基準能提供基於仿真相應的不同的激發向量，以便於測試基準功能作為一個被測試系統環境的動作模型正常工作。 TestBencher Pro是一個優秀的測試大規模FPGA 和 ASIC設計的工具。

2.BugHunter Pro

- 一款用於 Verilog，VHDL 和 C++ 模擬器的圖形調試系統。BugHunter 支持所有主要的 HDL 模擬器，其主要功能包括：單一步驟調試、波形數據串流、項目管理等。可以偵測程序與幫助系統的交互過程從而發現問題所在。

3.VeriLogger Pro

- Verilog模擬器軟件。提供一個綜合了傳統 Verilog模擬器所有特徵的仿真環境，它具有強大的圖形測試矢量產生器。VeriLogger 的快速模型測試使用戶可以對設計中的每個模型進行「自底向上」的測試。

4.Waveformer Pro

- 由時序圖分析器、模擬器和測試向量生成器組合而成。它為設計人員提供了一個集成的環境，可以用來開發數字和模擬線路的時序圖。
Waveformer Pro可以由**Spice**模擬器，**Mathematica**，或**Matlab**等工具輸入模擬數據。還可以利用**Waveformer Pro** 內部擁有的波形函數直接生成模擬信號；或者將從總線上得到的數字信號變換成模擬信號。還可以將模擬信號波形逐段線性地變換成**Spice**模擬器的電壓信號源。

5.SimuTAG

- 一款功能強大和實時地查證出 **FPGAs** 的錯誤的系統。可經由比較**FPGA** 功能規則反證**RTL**樣式模擬規則，迅速發現邏輯和綜合錯誤。

6.PinPort

- 提供數字硬件與Verilog 或C++的虛擬接口環境。

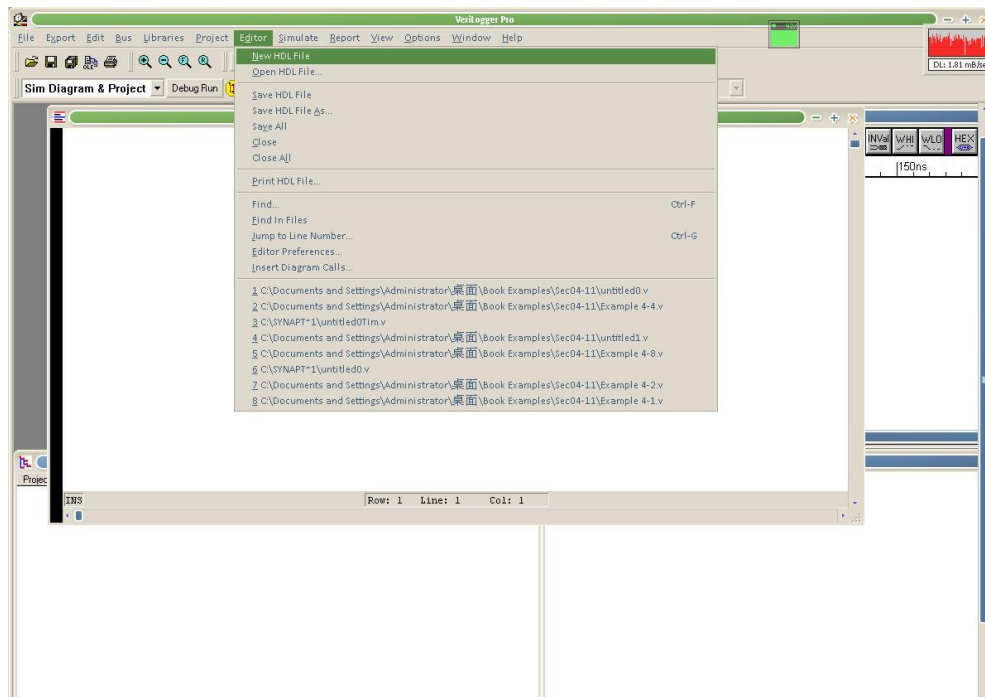
- 7.DataSheet Pro：時間表文件夾管理工具。
- 8.Timing Diagrammer Pro：時間表設計工具。
- 9.Gigawave Viewer：波形觀測器。

開啓SynaptiCAD



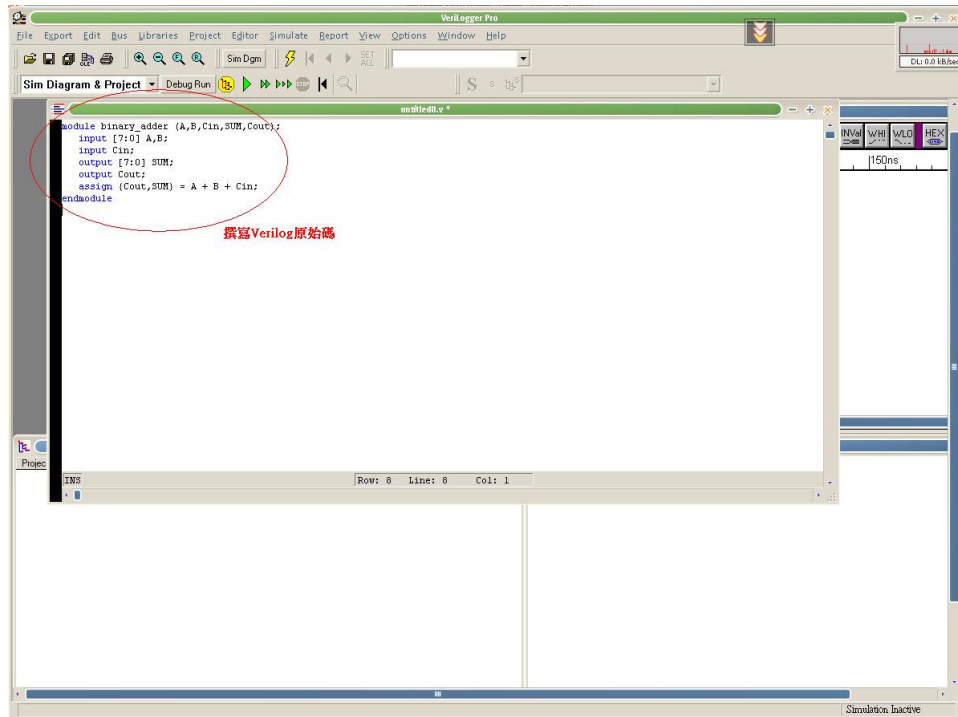
- 依照光碟步驟將**SynaptiCAD**安裝完成後，開啓**SynaptiCAD**中的Verilogger Pro來編譯已經撰寫好的*.v檔

新增Verilog檔案



Editor->New HDL File 新增一個Verilog檔案

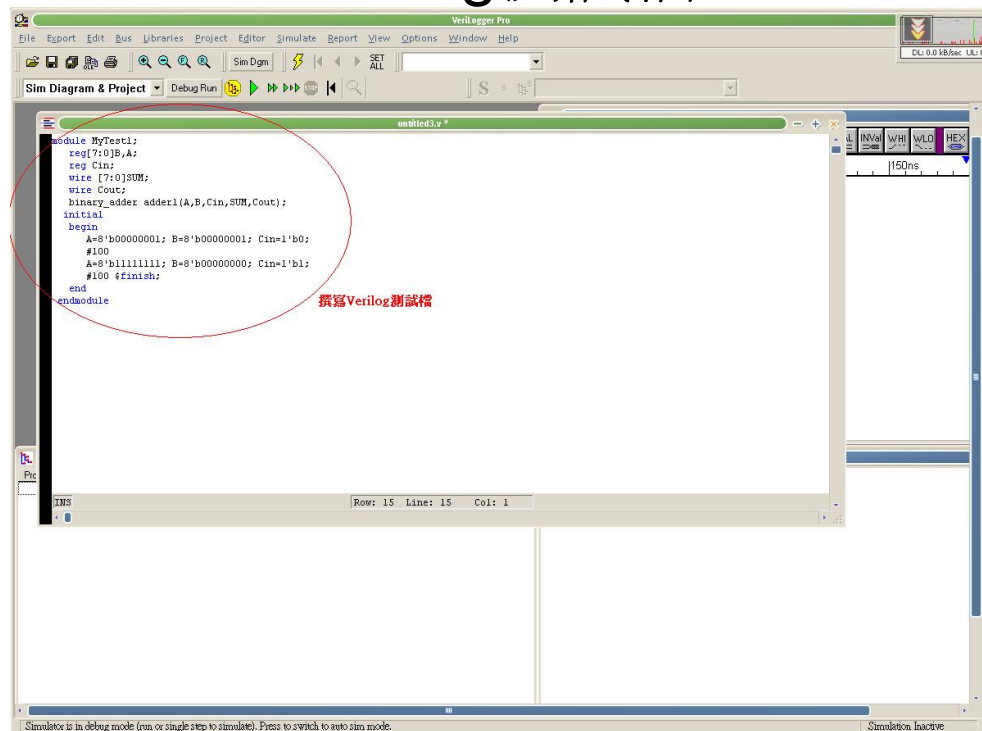
撰寫Verilog



撰寫Verilog原始碼

在視窗中撰寫Verilog原始碼

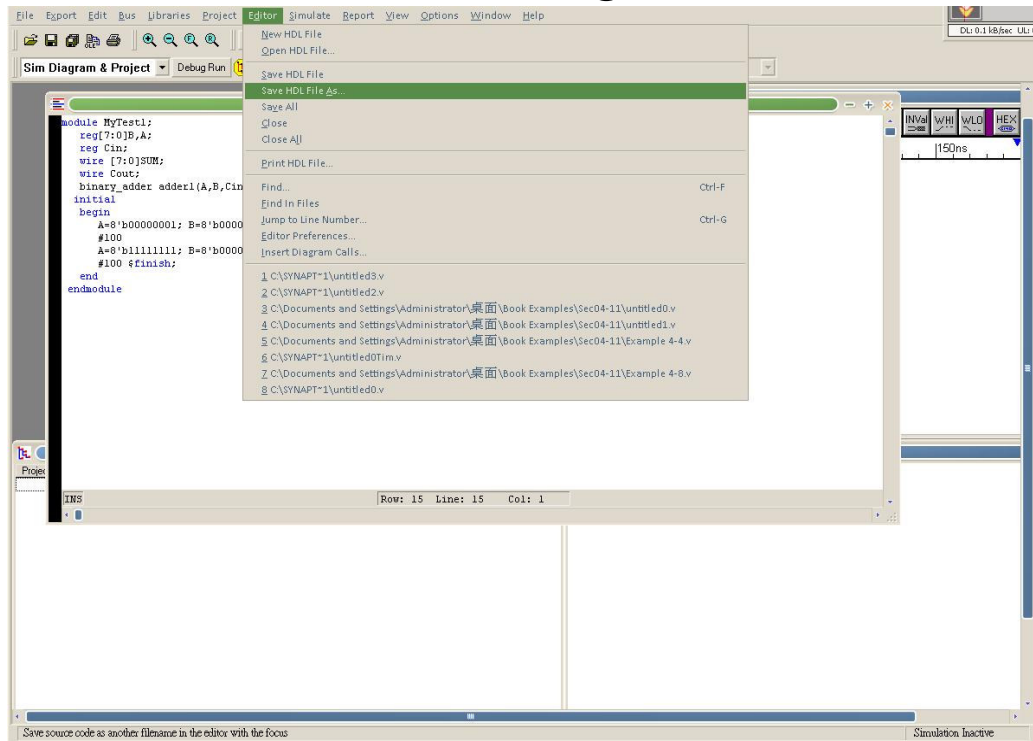
Verilog測試檔



撰寫Verilog測試檔

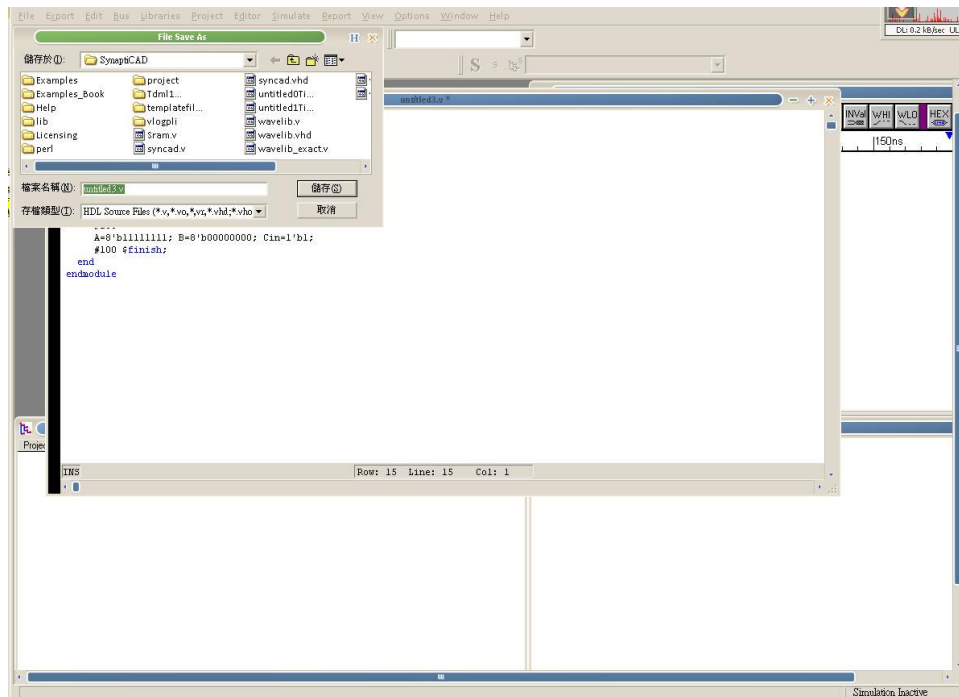
和上一步驟相同，新增一個HDL檔案後撰寫Verilog的測試檔

儲存Verilog檔案



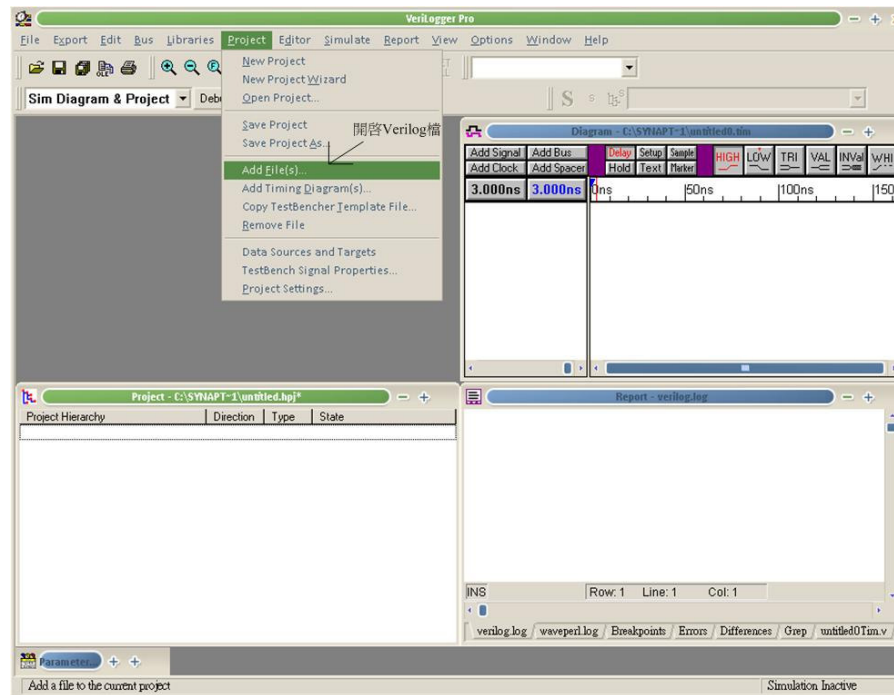
Editor->Save HDL File As 把撰寫完成的Verilog檔案和測試檔儲存

儲存檔名



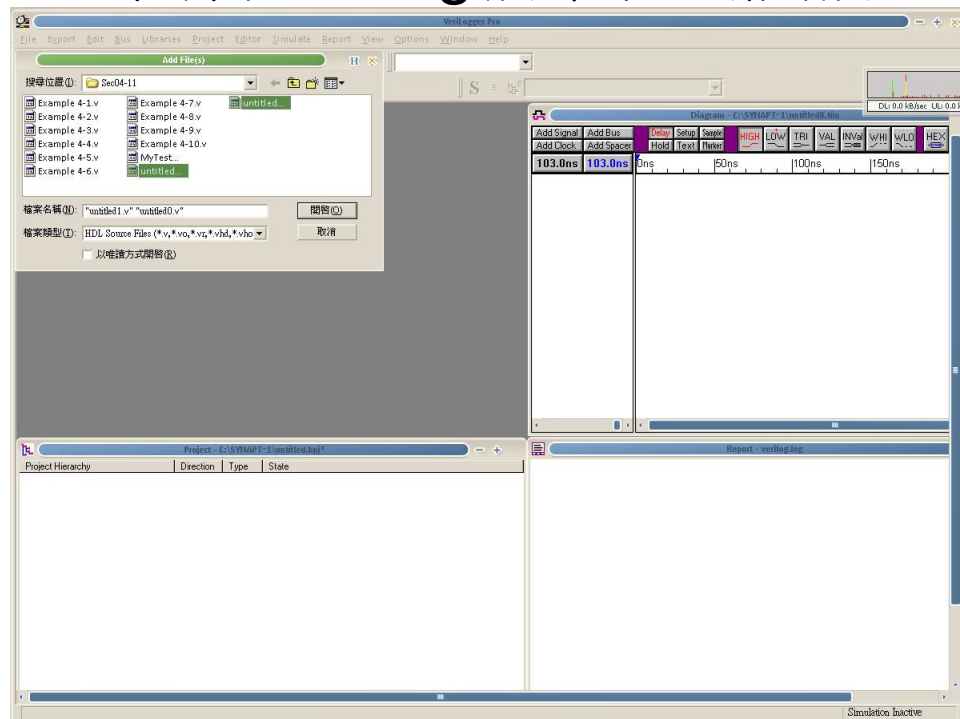
存好的Verilog檔案副檔名為 .v

在專案中開啓Verilog檔案



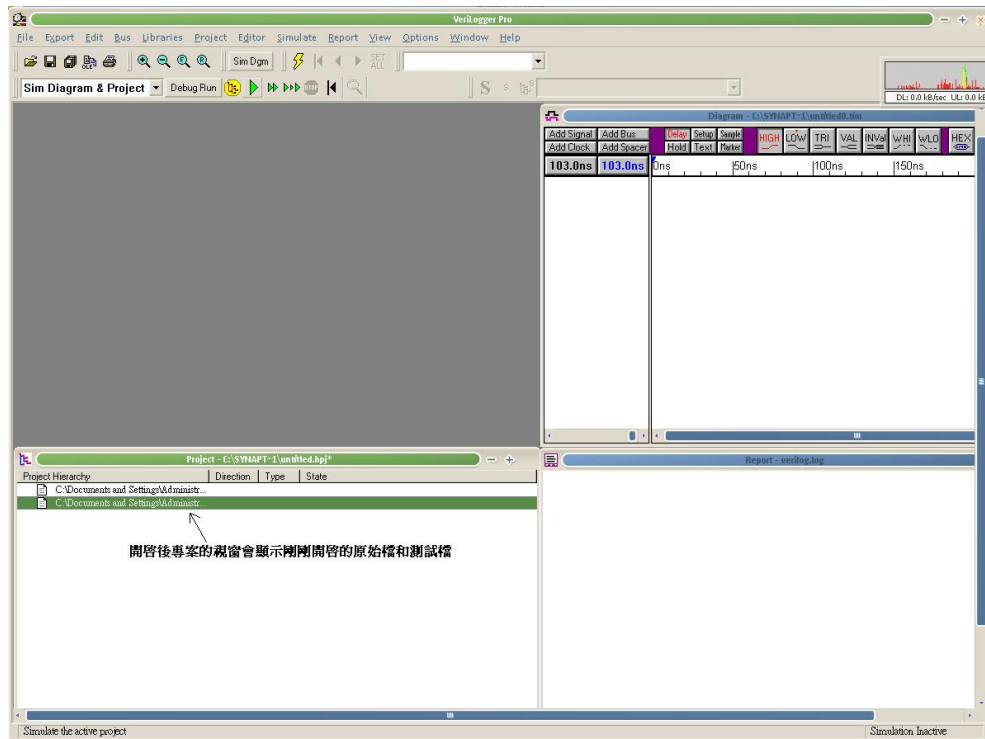
Project->Add File(s)... 在專案中開啓剛撰寫完成的Verilog檔案

開啓Verilog檔案和測試檔



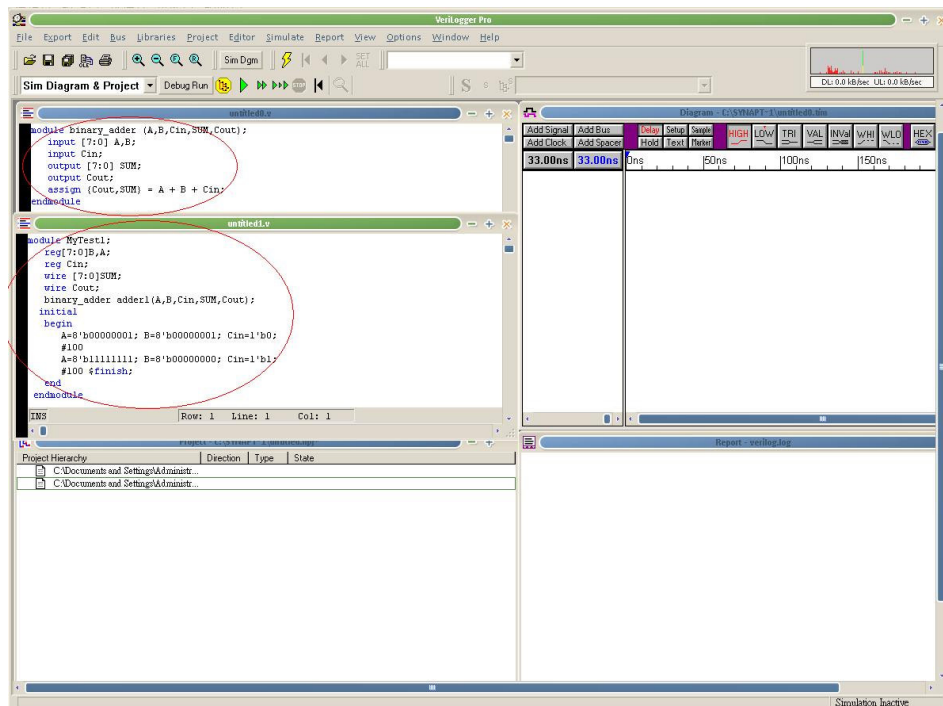
找到剛剛撰寫的Verilog檔案與測試檔路徑並開啓

專案視窗顯示



開啓後專案的視窗會出現剛開啓的檔案，點擊滑鼠左鍵兩下可看到Verilog的原始碼

原始碼視窗顯示



點擊專案檔裡的檔案兩下之後將出現紅色圈圈部分為Verilog原始碼視窗

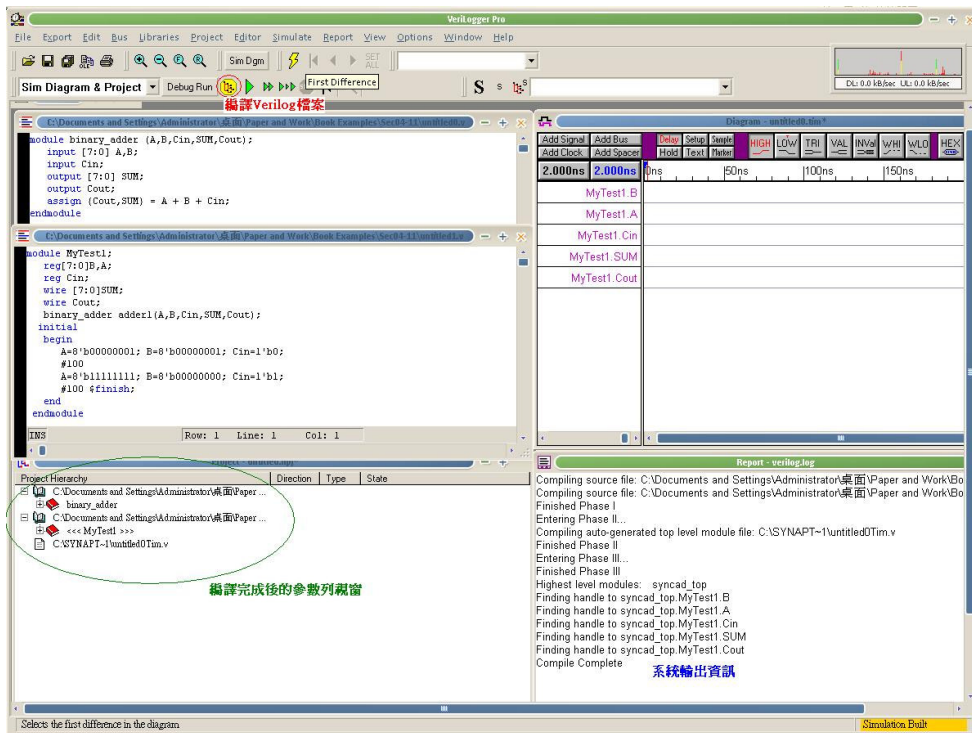
Verilog原始檔

```
module binary_adder (A,B,Cin,SUM,Cout);  
    input [7:0] A,B;  
    input Cin;  
    output [7:0] SUM;  
    output Cout;  
    assign {Cout,SUM} = A + B + Cin;  
endmodule
```

Verilog測試檔

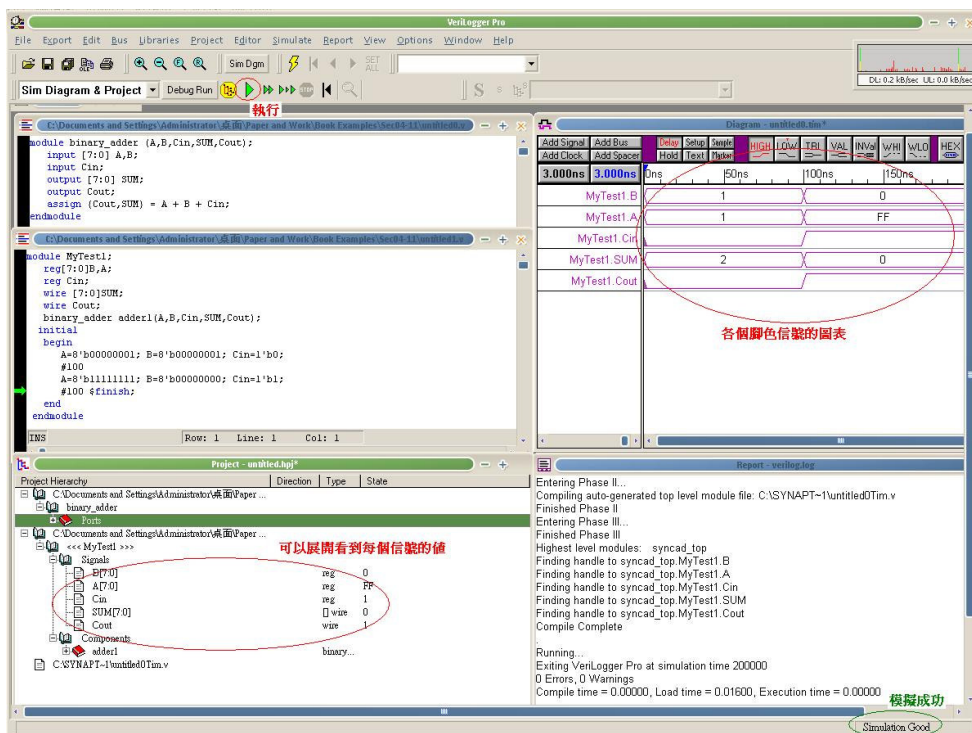
```
module MyTest1;  
    reg[7:0]B,A;  
    reg Cin;  
    wire [7:0]SUM;  
    wire Cout;  
    binary_adder adder1(A,B,Cin,SUM,Cout);  
    initial  
    begin  
        A=8'b00000001; B=8'b00000001; Cin=1'b0;  
        #100  
        A=8'b11111111; B=8'b00000000; Cin=1'b1;  
        #100 $finish;  
    end  
endmodule
```

編譯Verilog檔案



按下編譯鍵，編譯完成後會出現參數列和系統輸出資訊

執行後腳位圖表



編譯之後執行程式，會出現各信號腳位的圖表和數值